

FIG. 1 (PRIOR ART)

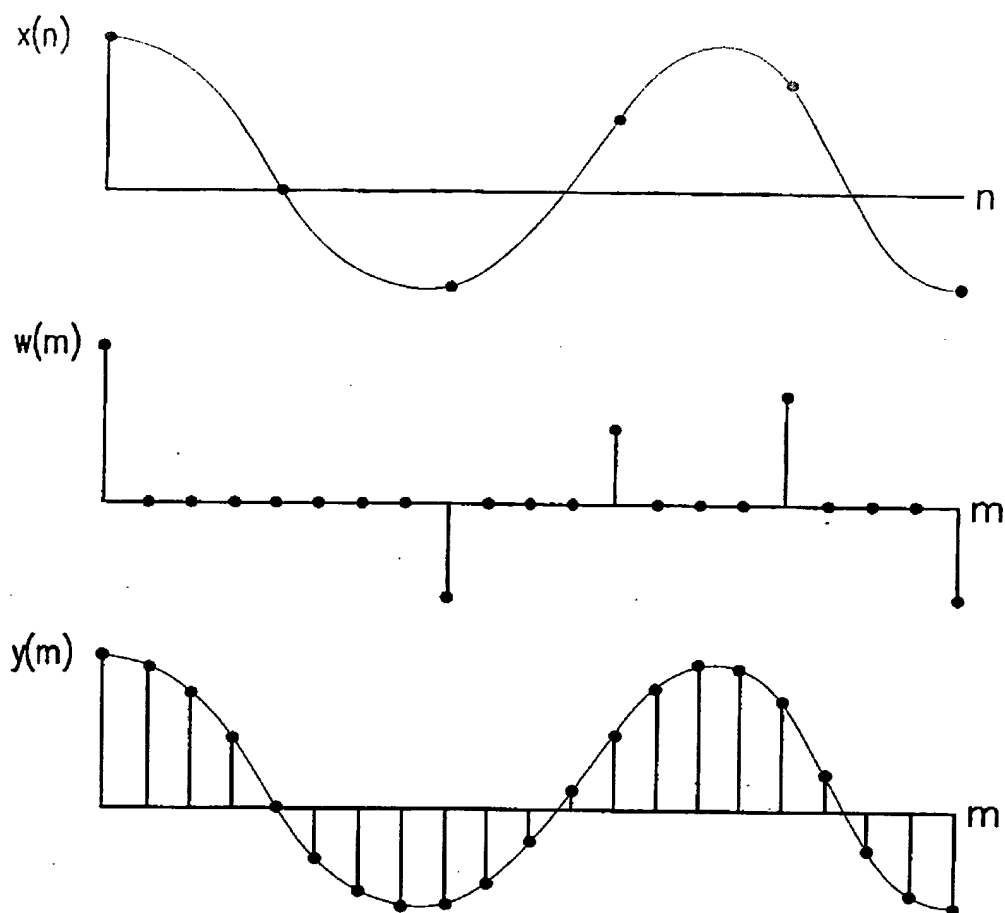


FIG. 2 (PRIOR ART)

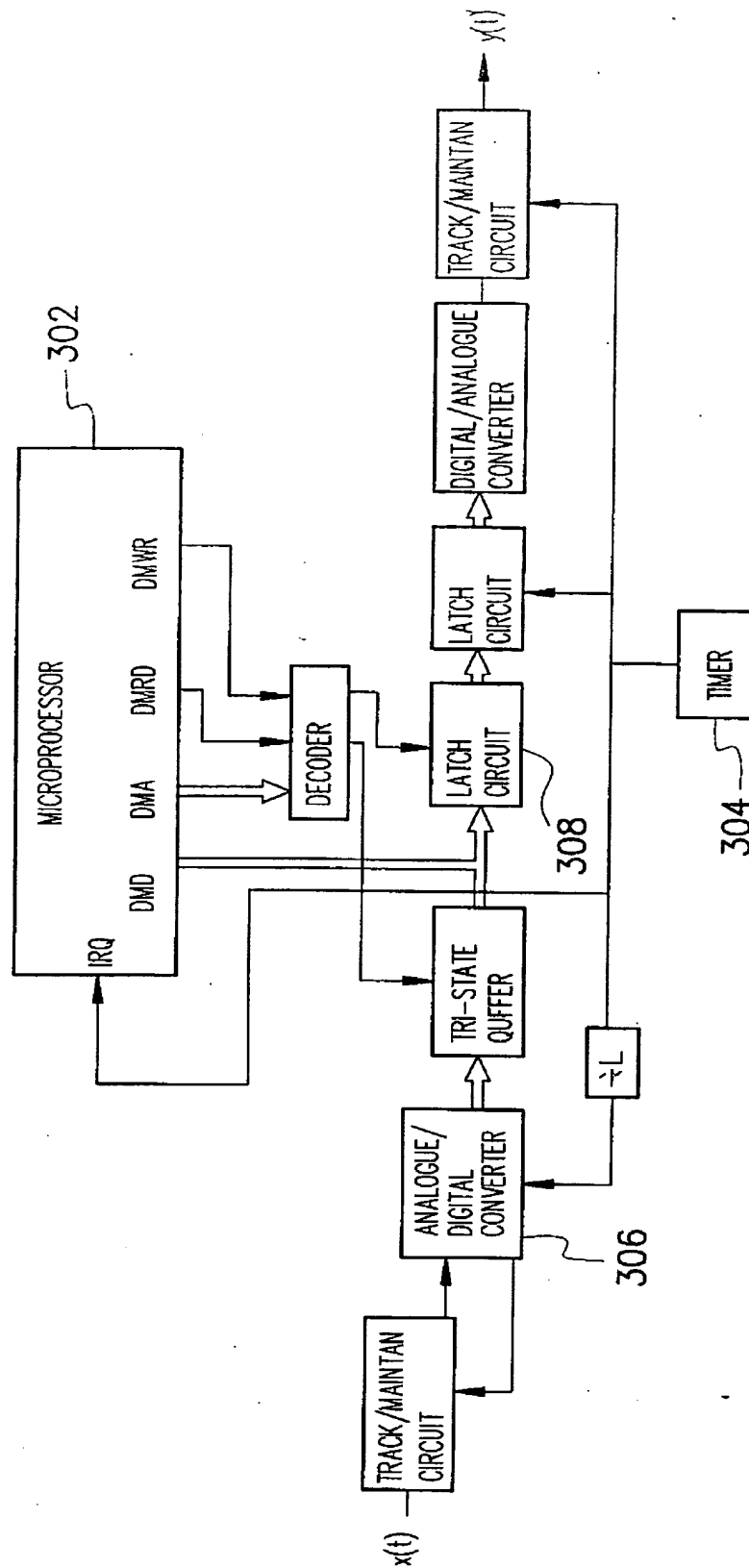


FIG. 3 (PRIOR ART)

```
{ INTERPOLATE.dsp
```

Real time Direct Form Filter, N taps, uses an efficient algorithm to interpolate by L for an increase of L times the input sample rate. A restriction on the number of taps that N/L be integer.

```
INPUT: adc
```

```
OUTPUT: dac
```

```
}
```

```
MODULE/RAM/ABS=0 interpolate;
```

```
.CONST          N=300;
```

```
.CONST          L=4;           { interpolate by factor of L }
```

```
.CONST          NoverL=75;
```

```
.VAR/PM/RAM/CIRC coef[N];
```

```
.VAR/PM/RAM/CIRC data[NoverL];
```

```
.VAR/PM/RAM/      counter;
```

```
.PORT           adc;
```

```
.PORT           adc;
```

```
.INIT           coef: <coef.dat>;
```

```
RTI;           {interrupt 0 }
```

```
RTI;           {interrupt-1 }
```

```
RTI;           {interrupt 2 }
```

```
JUMP sample;   {interrupt 3 at (L*input rate) }
```

```
initialize:    IMASK=b#0000; {disable all interrupts}
               ICNTL=b#01111; {edge sensitive interrupts}
               SI=1;          {set interpolate counter to 1}
               DM(counter0=SI; {for first data sample}
               I4=~coef;      {setup a circular buffer in PM}
               L4=%coef;
```

(listing continues on next page)

FIG. 4 (PRIOR ART)

```

M4=L;                {modifier for coef is L}
M5=-1;               {modifier to shift coef back -1}
IO=^data;            {setup circular buffer in DM}
L0=%data;
M0=1
IMASK=B#1000;        {enable interrupt 3}
wait_interrupt: JUMP wait_interrupt;{infinte wait loop}
{ _____ Interpolate _____ }
sample:  MODIFT(I4,M5);    {shifts coef pointer back by -1}
          AYO=DM(counter);
          AR=AYO-1;        {decrement and update counter}
          DM(counter)=AR;
          IF NE JUMP do_fir; {test ant input if L times}

{ ____ input data sample, code executed at the sample rate ____ }
do_input:  AYO=DM(adc);    {input data sample}
          DM(IIO,M0)=AYO;  {update delay line wiht newest}
          MODIFY(I4,M4);   {shifts coef pointer up by L}
          DM(counter)=M4;  {reset counter to L}

{ ____ filter pass, occurs at L times the input sample rate ____ }
do_fir:   CNTR=NOVERL -1;  {N/L since round on last tap}
          MR=0, MXO=DM(IO,M0); MYO=PM(I4,M4);
          DO taploop UNTLL CE; {N/L-1 taps of FIR}
taploop:  MR=MR+MXO*MXO(SS), MXO=DM(IO,M0), MYO=PM(I4,M4);
          IF MV SAT MR;    {saturate result if overflowed}
          DM(dac)=MR1;    {output sample}
          RTI;
ENDMOD:

```

FIG. 4 (PRIOR ART)